3.0 USER-COMPUTER INTERACTION

3.1 INPUT FOCUS

3.1.1 Focus Models

Two modes, explicit and implicit, are available for assigning input focus to a window. In explicit mode, focus can be moved among windows either with the pointing device or from the keyboard, and the keyboard can be used for navigation among the components in the window with focus. In implicit mode, focus moves with the pointer and cannot be controlled from the keyboard.

Regardless of the focus policy in effect, only one window on the screen has input focus at any time; within that window, only one object at a time has focus. Explicit focus is the default for the DII. However, systems may implement implicit focus or provide users with the flexibility to select either focus policy (see section 7.1.6). As a result, the application must be able to support both explicit and implicit focus (e.g., ensure that application-unique features designed for use with implicit focus also behave correctly if explicit focus is selected, and vice versa). Given that explicit focus is the default, the remainder of this document describes user interaction under this focus policy.

3.1.2 Assigning Focus with the Pointing Device

Users assign input focus by moving the pointer into a window or icon and clicking BSelect. The window with focus is indicated by a change in the color of the window frame.

3.1.3 Assigning Focus with the Keyboard

<Ctrl><Esc> displays a window listing currently running applications, and <Alt><Esc> (and <Alt><Shift><Esc> in Windows) moves focus between applications. <Alt><Tab> and <Alt><Shift><Tab> move focus forward and backward through the window families in the application, and <Alt><F6> and <Alt><Shift><F6> move focus forward and backward through the windows within a family.

> Windows Only: <Ctrl><F6> and <Ctrl><Shift><F6> move focus forward and backward through the document windows in the application.

<F6> (and <Shift><F6> in Windows) move focus between panes in the active window.

3.2 NAVIGATION

3.2.1 Pointing Device Navigation

Navigation refers to how users move the pointer and location cursor on the screen. In pointing device navigation, placing the pointer on an object and clicking BSelect moves the location cursor to the object and gives it focus; the object is also selected or activated (and it is highlighted). The highlighting remains visible only while the window in which the object appears has focus. Clicking BSelect does not move focus to an object that is not selectable and cannot accept input. Holding down <Ctrl> while clicking BSelect on an object moves keyboard focus without selecting or activating the object.

If autoscrolling is available for a scrollable control such as text or a list box, placing the pointer on the control, pressing BSelect, and dragging the pointer outside the control scrolls the control in the direction of the pointer. Releasing BSelect stops autoscrolling.

3.2.2 Keyboard Navigation

3.2.2.1 The Location Cursor

The object that has focus in a window is identified by a location cursor. Only one location cursor appears in a window at any time. When focus is explicit, movement of the location cursor is controlled from the keyboard; movement of the pointing device is mapped to the pointer and does not affect the position of the location cursor on the screen.

The shape of the location cursor depends on the type of object with keyboard focus. Figures 3-1 and 3-2 show the location cursors used in Motif and Windows.

Motif Only: The box cursor is used in controls such as radio buttons, check buttons, push buttons, and text fields; the box is displayed outside the control (e.g., encloses the indicator and label in radio and check buttons). The shadow cursor[1] is used in menus (i.e., objects whose outline is not normally shown). The item cursor is used in lists, while the fill cursor is used in very small objects such as sashes in paned windows. The text cursor is shown in the text object (e.g., text field) with keyboard focus and indicates where text typed by users will be displayed.

Windows Only: The dotted box cursor is used in controls such as radio buttons, check buttons, push buttons, and list boxes; the dotted box encloses the label of the control. Highlighting is used as the location cursor in menus. When a text field initially receives focus, it highlights and the text cursor is displayed in the field. Clicking BSelect on the text removes the highlight and places the text cursor at the click location.

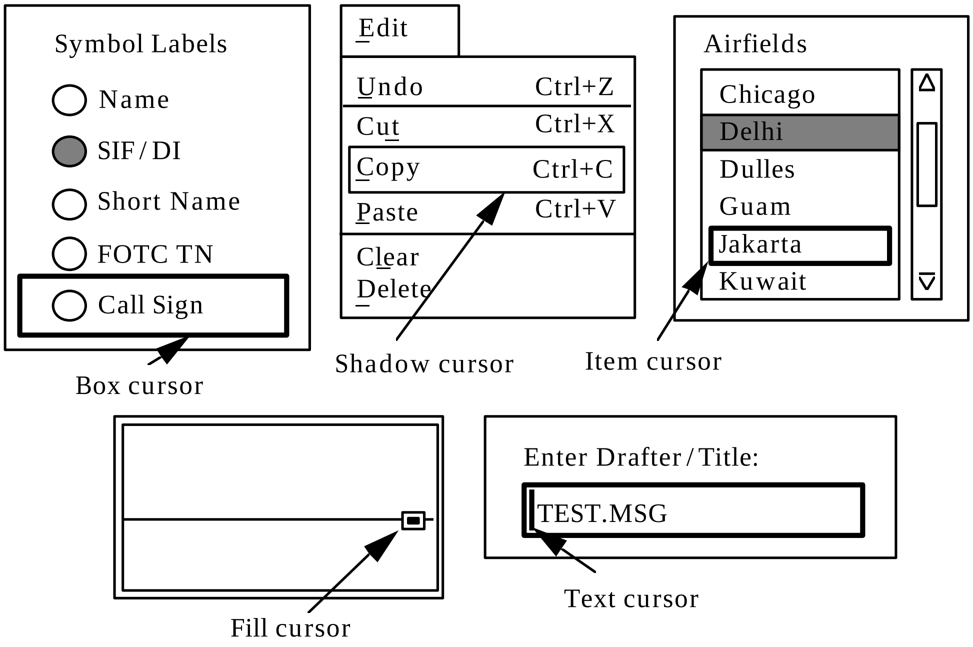

Figure 3-1. Example location cursors in Motif.

[1] The Motif Style Guide refers to this cursor shape as the outline highlight location cursor. The specifications presented here refer to this shape as the shadow cursor to minimize confusion with the highlight which is the change in color (e.g., reverse video) that occurs when an object is selected.

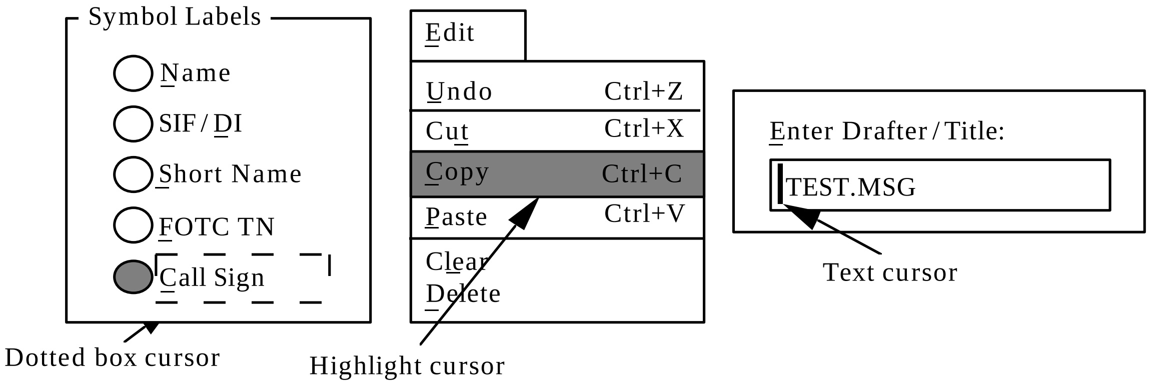

Figure 3-2. Example location cursors in Windows.

The shape of the text cursor is a vertical bar (|) in insert mode and and a shaded rectangle in replace mode.[2] The text cursor flashes when the object containing the cursor has keyboard focus. The flash rate is 2-5 Hz. The text cursor is easy to see whenever it appears in a text area. Visibility can be improved by changes in bolding, contrast, and size of the text cursor.

The application uses existing cursor shapes to indicate keyboard focus for any new objects for which a location cursor is not already available. New cursor shapes can be created when necessary to support the functionality provided by the application.

3.2.2.2 Navigation in Tab Groups

The content of a window is divided into tab groups in order to support keyboard navigation for controls. A tab group may contain a single control, such as a list box or text field, or a set of related controls, such as a set of radio buttons, check buttons, or push buttons. Navigation keys move the location cursor between tab groups in a window and between controls within a group.

<Tab> (and <Ctrl><Tab> in Motif) moves the location cursor to the next tab group in a window, and <Shift><Tab> (and <Ctrl><Shift><Tab> in Motif) moves the cursor to the previous group. If none of the controls in a tab group can have keyboard focus (e.g., none is available for selection or activation), then <Tab> (and <Ctrl><Tab> in Motif) skips the group.

<Up>, <Down>, <Left>, and <Right> move the location cursor between the controls within the tab group with keyboard focus. The arrow keys move the location cursor one increment at a time (e.g., to the next line in text, to the next item in a list box). <Ctrl> in combination with the arrow keys increases the size of the increment (e.g., moves the text cursor to the next word rather than the next character in text). If a graphics-like object uses a positional cursor, the arrow keys move the cursor one unit (e.g., one pixel) at a time in the direction indicated by the arrow.

<Home> moves the location cursor to the leftmost element in the data, and <End> moves the cursor to the rightmost element. <Ctrl><Home> moves the location cursor to the top leftmost (i.e., beginning) element in the data, and <Ctrl><End> to the bottom rightmost (i.e., end) element. In controls such as text fields and list boxes that are scrollable, <PageUp>, <PageDown>, <Ctrl><PageUp>, and <Ctrl><PageDown> scroll the elements in the control up, down, left, or right one page minus one unit of information (e.g. one line of text) at a time, as appropriate.

[2] Previous versions of Motif use the same text cursor shape in both insert and replace mode.

Windows Only: <ScrollLock> toggles the scroll lock mode; when the mode is enabled, the navigation keys scroll data without moving the location cursor or affecting existing selections.

3.2.2.3 Location Cursor Behavior During Navigation

The application defines tab groups within each window to support efficient navigation among sets of related controls. The identification of tab groups is based on the order in which users are expected to interact with the controls in the window. This sequence is left to right, top to bottom in the window. For example, in figure 3-3, users tab from the set of radio buttons in Text Size to the text field in Text Size, then to the set of radio buttons in Symbol Size, and finally to the push buttons at the bottom of the window.

Motif Only: Text fields may be considered as individual tab groups, with <Tab> used for navigation between groups, or sets of single-line text fields may be treated as a single tab group, with the arrow keys used to navigate among them. Similarly, push buttons can be defined as individual tab groups or treated as a single tab group.[3]

Windows Only: Push buttons are defined as individual tab groups.

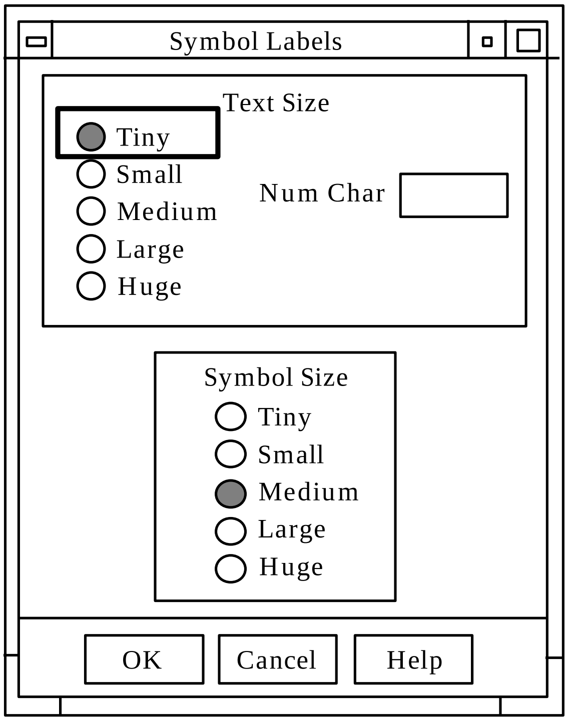

Figure 3-3. Example symbol label window in Motif.

When a window is first displayed, the location cursor is placed on the control with which users are expected to interact first. This control is usually the top leftmost one on which the location cursor can be positioned. For example, when the window shown in figure 3-3 is displayed, the location cursor is placed on the Tiny radio button in the Text Size tab group. When a window regains input focus, the location cursor is placed on the control that last had focus, provided that the control remains available

[3] In previous versions of Motif, push buttons were defined as a single tab group; CDE provides the flexibility to use either <Tab> or the arrow keys to navigate within a push button group.

for selection; otherwise, the location cursor is placed on the control users are most likely to select in the window.

The direction of cursor movement within the window is from upper left to lower right unless the object is scrollable; the location cursor wraps between the first and last tab groups in the window. When the location cursor moves into a tab group, the cursor is placed on the first available control in the group. The location cursor skips a tab group if none of the controls can have keyboard focus. In scrollable controls, keyboard focus remains on the element where it was positioned before the scrolling operation began even though the location cursor may no longer be in view. However, any keyboard action that moves the location cursor or makes a modification at the cursor location scrolls the control so that the cursor is visible.

Once the location cursor is positioned within a tab group, the arrow keys move the cursor between the available controls within the group. For example, pressing <Down> in the Symbol Size tab group moves the location cursor from Tiny to Small to Medium and so on. The location cursor is always visible as it moves within a window; i.e., there are no “invisible” tab groups which cause the location cursor to temporarily disappear as users navigate within the window. The movement of the location cursor is controlled from the keyboard; the position of the location cursor is not affected by movement of the pointer within the window. For example, the location cursor does not appear on a control within a tab group when the pointer moves into the group, and then disappear when the pointer moves out of the group.

Moving the location cursor to a control does not alter the size or position of the control. Motif and Windows differ with regard the effect of keyboard navigation on the state of the control.

Motif Only: Moving the location cursor does not change the state of the control (i.e.., does not move the highlight to the control).

Windows Only: Keyboard navigation changes select state, unless Scroll Lock mode is enabled.

3.2.2.4 Text Cursor Behavior During Navigation

To assign keyboard focus to an area where text entry is possible, users move the pointer into the area and click BSelect. If the area is empty, the text cursor appears at the beginning (i.e., top leftmost part) of the area. If the area contains text, the text cursor appears between the characters that are under the pointer. If the pointer is positioned beyond the end of the text, the text cursor appears following the final text character.

<Tab> moves focus to a text field from the keyboard, highlights any text present in the field, and displays the text cursor in the field.

Motif Only: When users move the text cursor between text fields, the cursor appears at the beginning of whatever text appears in the field.

Windows Only: When users move the text cursor between text fields, the cursor appears at the end of whatever text appears in the field.

The arrow keys move the text cursor one character or one line in the direction indicated by the arrow.

The appearance of the text cursor changes when the area in which it is displayed loses focus. The text cursor disappears when the area loses focus and reappears when it regains focus. If the cursor is not removed, then it is grayed out and stops flashing when the area loses focus, and then returns to normal appearance and resumes flashing when the area regains focus. If the text cursor is removed from view when an area loses focus, the cursor re-appears at the same location when the area regains focus.

Users can place the text cursor only in areas where text entry is possible, and cannot place the cursor in areas where text entry is not possible (e.g. noneditable text fields). Text entry is possible only after the text cursor is visible at a location that can accept text entry. Text entry is not possible (i.e., is not accepted by the application) when the text cursor is not visible.

3.3 SELECTION

3.3.1 Pointing Device Selection Methods

Selection refers to the ways that users identify the elements of the interface with which they want to interact. Table 3-1 lists the methods the application is to use for selecting elements in a collection using the pointing device. In some cases, the methods vary slightly depending on whether the element is part of a one-dimensional collection such as text or a list box (i.e., where the elements can be treated as if they are in a line) or a two-dimensional collection such as a group of graphics objects (i.e., where an element has both a vertical and horizontal position within the collection). The selection methods listed in table 3-1 also apply to performing margin selection (i.e., for lines, rows, and/or columns of data) in Windows.

> Motif Only: When an integrated Select/Transfer model is implemented, users can choose to map the Adjust or Transfer function (see section 2.1.3) to button 2 on the pointing device. As a result, users can perform the Adjust function in making a selection using either BSelect (as was true with previous versions of Motif) or BAdjust (when the Adjust function is assigned to button 2). If the application uses CDE Motif, it supports both alternatives for performing selection.

> Windows Only: Windows supports integrated selection and transfer using BSelect (see section 2.1.3). To distinguish between these operations, drag handles are displayed on or around an object when it is selected, and the pointer shape changes when users initiate a drag operation (see section 3.5.1 on drag transfer).

Table 3-1. Pointing device selection methods in Motif and Windows.

Single Selection:

To select one element, place the pointer on an unselected element and click BSelect. The location cursor moves to the element, it is selected, and any element that was selected is deselected.

To browse the elements in a collection, press BSelect and drag the pointer over the elements in the collection. When the pointer is on an element, it is selected, and any other element that was selected is deselected. If BSelect is released on an element, it remains selected, and the location cursor moves to the element.

Motif Only: To select multiple elements at a time, place the pointer on the first element and click BSelect or BAdjust to select it. If previously unselected, the element is selected; if previously selected, the element is deselected. In both cases, the location cursor moves to the element, and any element that was selected remain selected.

Range Selection:

To perform a range selection, press BSelect on the first element, drag the pointer until it is on the last element in the range, and release the button to complete the selection.[4] In one-dimensional collections: The elements within the range are selected as the pointer is dragged over them. In two-dimensional collections: As the pointer is dragged through the elements, a bounding box is displayed outlining the elements being selected. When BSelect is released, the box disappears and the elements that were in the box are selected.

To extend a range selection, place the pointer on the last element to be added to the selection, then press <Shift> and click BSelect (or click BAdjust in Motif, or press <Shift><Ctrl> and click BSelect in Windows). The elements in the revised selection range (defined from the original anchor to the current pointer position in one-dimensional collections, and defined by the diagonal from the anchor to the current pointer position in two-dimensional collections) are selected, and any elements removed from the selection are deselected.

Discontiguous Selection:

To perform a discontiguous selection, place the pointer on the discontiguous element, then press <Ctrl> and click BSelect (or click BAdjust in Motif). If previously unselected, the element is selected; if previously selected, the element is deselected. The other elements in the selection remain selected. The location cursor moves to the element, and it becomes the anchor for the selection.

To extend a discontiguous selection, press <Ctrl> and BSelect (or press BAdjust in Motif) to set the anchor for the range. Drag the pointer until it is on the last element in the range, and release the button to complete the selection. In one-dimensional collections: If previously unselected, the elements in the range are selected as the pointer is dragged over them; if previously selected, the elements in the range are deselected. In two-dimensional collections: As the pointer is dragged through the elements, a bounding box is displayed outlining the elements. When BSelect (or BAdjust in Motif) is released, the box disappears. If previously unselected, the elements in the box are selected; if previously selected, the elements in the box are deselected.

[4] In Motif, if BSelect is released before the drag threshold is exceeded, the element under the pointer is selected, and any other element that was selected is deselected. If the drag threshold is exceeded, a drag operation is initiated, and any previously selected element remains selected.

The selection method(s) available in a collection match the type of selection that users are expected to make in the collection. For example, if users are expected to select a single item in a list box, the selection method available in the list allows them to select only one item; they cannot select multiple items or deselect all items in the list. Deselection affects only the select state of the elements and does not delete any of the elements in the collection.

Windows Only: <F8> toggles Extend mode; when this mode is enabled, selection behaves as if <Shift> is locked down for all direction keys and pointing device actions. Windows supports selection modes but recommends that they be used only in special contexts. If available, they supplement standard selection methods and include a visual cue when the mode is enabled (e.g., mode information is displayed in the message bar of the window).

3.3.2 Keyboard Selection Methods

Two modes, normal and add, are available for selecting the elements in a collection using the keyboard. In normal mode, the location cursor and the highlight (indicating the current selection) move together when the arrows keys are used. In add mode, the location cursor moves independently of the highlight. The shape of the location cursor is a solid rectangle in normal mode and a dotted rectangle in add mode. <Shift><F8> toggles between the two modes. Toggling between modes does not affect the select state of objects previously selected using the pointing device.

Motif and Windows differ in their use of these modes in making selections. Table 3-2 lists the keyboard selection methods to be used in Motif applications, while table 3-3 lists the methods to be used in Windows applications.

Motif Only: Add mode is used in single and multiple selection, and normal mode in browse and range selection. Add mode may also be used in range selection. Both add and normal modes are used in discontiguous selection. Because <Space> and <Shift><Space> are mapped to text entry actions (e.g., insert a space character in text), the application is to use the alternate set of key bindings provided by Motif for performing range selection in text collections (see appendix A).

Windows Only: Normal and add mode are used in single selection, and normal mode in range selection. Both add and normal modes are used in discontiguous selection.

Table 3-2. Keyboard selection methods in Motif.

Single Selection:

To select one element, use the arrows keys to move the location cursor to an unselected element. The location cursor moves independently of the highlight. <Space> or <Select> selects the element and moves the highlight to it, and any element that was selected is deselected.

To browse the elements in a collection, use the arrow keys to move the location cursor to an element and select it; any element that was selected is deselected.

To select multiple elements one at a time, use the arrow keys to move the location cursor to an element. The location cursor moves independently of the highlight. <Space> or <Select> selects an unselected element or deselects a selected element.

Range Selection:

To perform a range selection using normal mode, use the arrow keys to move the location cursor to the first element in the range. Press <Space> or <Select> to set the anchor for the selection. The element is selected, and any element that was selected is deselected. Use the arrow keys to move the location cursor and highlight to the last element in the range. <Shift><Space> or <Shift><Select> extends the selection from the anchor to the location cursor. The location cursor remains on the last element in the range.

To perform a range selection using add mode, use the arrow keys to move the location cursor to the first element in the range. Press <Space> or <Select> to set the anchor for the selection. The element is selected, and any other element that was selected is deselected. Use the arrow keys to move the location cursor to the last element in the range; the highlight remains on the anchor element. <Shift><Space> or <Shift><Select> extends the selection from the anchor to the location cursor and selects all of the elements in the range. The location cursor remains on the last element in the range.

To extend a range selection using normal or add mode, use the arrow keys to position the location cursor on the last element in the selection. Press <Shift> and use the arrow keys to move the location cursor to the other elements to include in/remove from the selection. Elements being added to the selection highlight as the location cursor moves to them; elements being removed from the selection return to normal appearance.

Discontiguous Selection:

Select the elements in the contiguous part of the collection using normal mode (see above). Press <Shift><F8> to toggle to add mode. Select the element(s) in the discontiguous part of the collection using add mode (see above). Any elements that were selected using normal mode remain selected. Press <Shift><F8> to toggle out of add mode.

Table 3-3. Keyboard selection methods in Windows.

Single Selection:

To select a single element using normal mode, use the arrow keys to move the location cursor to an element. The highlight moves with the location cursor, and any element that was selected is deselected.

To select a single element using add mode, use the arrows keys to move the location cursor to an unselected element. The location cursor moves independently of the highlight. <Space> selects the element, moves the highlight to it, and any element that was selected is deselected.

Range Selection:

Use the arrow keys to move the location cursor to the first element in the range. The anchor for the selection is set to this element. Press <Shift> and any navigation key(s) to move the location cursor to the last element in the range. In one-dimensional collections: When these keys are pressed, the location cursor moves to the last element in the range and all elements in the range highlight. In two-dimensional collections: If the anchor element was selected, all elements in the range are selected when the keys are pressed; if the anchor element was deselected, all elements in the range are deselected.

Discontiguous Selection:

Select the elements in the contiguous part of the collection using normal mode (see above). Press <Shift><F8> to toggle to add mode. Press <Space> to toggle the select state of the element under the location cursor and set the anchor for the selection. Select the element(s) in the discontiguous part of the collection using add mode. Any elements that were previously selected remain selected. Press <Shift><F8> to toggle out of add mode.

<Backspace> and <Delete> delete the current selection.

Motif Only: <Ctrl></> is available, as appropriate, to select all of the elements in a collection and <Ctrl><\> to deselect all of the elements in a collection. In both cases, the location cursor remains at its current location.

3.4 ACTIVATION

3.4.1 Basic Activation

Activation refers to the ways users invoke the action associated with a window component or control. Placing the pointer on a button and clicking BSelect activates the button. Pressing <Space> (or <Select> in Motif) on the button with focus also activates the button. When BSelect is pressed on a button, the appearance of the button changes to indicate pending activation. This change is usually shown by highlighting the element (i.e., change in color, reverse video). If the pointer is moved off the button while BSelect is pressed, the button is not activated, and it returns to its normal appearance.

3.4.2 Mnemonics and Accelerators

A mnemonic is a single alphanumeric character in the label of a menu or control. In the window with input focus, pressing <Alt> and the mnemonic for an object moves the location cursor to the object and selects or activates it. If the location cursor is already on the object, pressing the mnemonic activates it. If the menu or control is unavailable (i.e., the label is dimmed), pressing the mnemonic has no effect (i.e., executes no action) and focus remains unchanged. An accelerator consists of one or more keys that execute an action regardless of the position of the location cursor when the accelerator is pressed. An accelerator is executed only when the window containing the accelerator has focus. The use of mnemonics and accelerators in menus and controls is addressed in section 5.5.5 and 8.1.2.12.

3.4.3 Default Activation

When a default action is assigned to an object, the action is executed with the pointing device by double clicking BSelect on the object. If the object is used for making selections (e.g., a list box), the element under the pointer is selected and then the default action is executed. In addition, after making a selection in a window, <Enter> (and <Ctrl><Return> in Motif) invokes the default action from the keyboard. If keyboard focus is on an object other than multi-line text, <Return> in Motif also executes the default action in a window. If a default action is available, users can reverse the effects of the action (e.g., by selecting an Undo option or a Cancel push button).

3.4.4 Expert Activation, Previewing, and Autorepeat (Motif Only)

If expert activation is implemented for an object, double clicking BSelect on the object performs the regular action of the object but in a more global manner. An expert action is available only in a group of push buttons or a group of radio buttons where one of the buttons is always selected; the action is available only as a short-cut to features provided elsewhere in the window. When keyboard focus is on a button used for expert activation, there is no default action available, unless the default and expert actions are the same.

If previewing is implemented, placing the pointer on a push button or radio or check button and pressing BSelect displays information describing what happens when the button is activated; the information is removed when BSelect is released. If previewing is available, the application provides a means for users to disable it.

If autorepeat is implemented, placing the pointer on a push button and holding down BSelect executes its action repeatedly at regular intervals; autorepeat stops when BSelect is released.

3.4.5 Cancel Activation

<Esc> (or <Cancel> in Motif) cancels the action being executed and returns the object to its state prior to the action. For example, <Cancel> in the dialog window with input focus is equivalent to selecting the Cancel push button in the window.

3.5 TRANSFER

3.5.1 Drag Transfer

3.5.1.1 Drag Transfer Operations

Drag transfer allows users to move, copy, and link objects by dragging them from one location to another. The application supports these drag transfer operations.

> Motif Only: Drag transfer is available for all objects that are represented as icons and for all elements that users can manipulate. Drag transfer provides redundant access to functionality available elsewhere in the window (e.g., through menus or push buttons).

As indicated in section 2.1.3, the preferred implementation in the DII is integrated Select and Transfer functions. Motif users are assumed to have a three-button pointing device and so will perform drag operations using BSelect and can also perform them using BTransfer (when the Transfer function is assigned to button 2),[5] while Windows users are assumed to have a two-button pointing device and so will perform drag operations using BSelect only.

Motif Only: Drag transfer is performed in the following manner:

To perform a drag move, users place the pointer on an object, then press <Shift> and drag the object using BSelect (or BTransfer). When the button is released, a copy of the object is pasted at the location of the pointer, and the original object is deleted.

To perform a drag copy, users place the pointer on an object, then press <Ctrl> and drag the object using BSelect (or BTransfer). When the button is released, a copy of the object is pasted at the location of the pointer, and the original object remains at its previous location.

To perform a drag link, users place the pointer on an object, then press <Ctrl><Shift> and drag the object using BSelect (or BTransfer). When the button is released, a link is created from the destination to the object at its original location.

The default drag operation if no modifier key is used is a move. BSelect always initiates a drag if the drag is started on a selected element; the drag starts once the drag threshold has been reached.

Windows Only: To perform a drag transfer, users place the pointer on an object and drag it to a new location using BSelect. When the button is released, a copy of the object is pasted at the location of the pointer, and the original object is deleted. How the operation is interpreted is determined by what the destination defines as the appropriate default operation. The most common default is a drag move, but the operation can also be interpreted as a drag copy or a drag link.

<Esc> (or <Cancel> in Motif) cancels a drag operation that is in progress and returns the object being dragged to its original location; releasing BSelect (or BTransfer in Motif) when not over a drop site also ends a drag operation.

Dragging a set of selected elements drags the entire selection. When a set of selected elements is moved within a collection, the elements remain selected after they have been moved.

Motif Only: Dragging an unselected element drag only the element and does not affect any other elements in the collection that may be selected. Dragging in a collection of overlapping elements occurs on the highest draggable element in the stacking order within the collection.

3.5.1.2 Feedback During Drag Transfer

During a drag operation, the pointer changes appearance (i.e., into a drag icon) to indicate that a drag is in progress.

Motif Only: The drag icon is composed of a source indicator, an operation indicator, and a state indicator.[6] The source indicator represents the type of object(s) being dragged. The operation indicator shows the type of drag operation being performed; an operation indicator is included in the drag icon if the operation is a copy or link but not if it is a move. The state indicator has

[5] In previous versions of Motif, the preferred implementation for drag transfer was to use BTransfer. CDE requires applications to support the use of BSelect to perform drag transfer operations.

[6] A set of standard graphics for drag icons will be included in a future version of this document.

an arrow shape (with a hotspot for positioning) when the drag icon is on a valid drop site, and a “cannot” shape when the icon is on an invalid drop site.

Windows Only: As the pointer moves, the object, its outline, or a representation of the object moves with the pointer.

The drop site changes appearance when the drag icon is placed on it. Options in Motif include showing a solid line around the site, a raised or lowered edge arond the site, or a pixmap drawn over the site. The preferred implementation for DII is a beveled edge that makes the drop site look recessed. Options in Windows include highlighting or other form of visual emphasis.

If a drag transfer is successful, the object is placed at the drop site and the drag icon is removed. If the transfer does not occur immediately, the pointer changes to a “busy” shape until the transfer is complete. If a drag transfer fails, the object remains at the drag source and the drag icon is removed.

Motif Only: The drag icon “melts into” the drop zone when dropped on a valid site and “snaps back” to the drag site when dropped on an invalid site. If the transfer fails, an error message is displayed that informs users why the failure occurred. If multiple elements are involved in a drag operation and the transfer is only partially successful, this feedback indicates which transfers succeeded and which failed. If the application provides help on a drag transfer, the dialog window providing this information contains a Cancel push button for cancelling the transfer operation.

3.5.1.3 Performance Guidelines (Motif Only)

The TED Style Guide provides the following performance guidelines with regard to executing drag transfer operations in Motif:

When BSelect is pressed, a drag transfer is initiated when the pointer has moved 10 pixels. When BTransfer is pressed, there is no drag threshold. A drag icon is displayed within 50 msec (maximum 70 msec) after the drag is initiated.

When the drag icon is moved over a drop zone, the drag icon or drop zone changes appearance within 50 msec (maximum 70 msec).

When the drag icon is dropped on a drop zone, feedback (either “melt into” or “snap back”) occurs within 50 msec (maximum 120 msec) of releasing the pointing device button. The visual effects last 200-350 msec (maximum 500 msec)

Data transfer is completed in 0.3-1.0 sec (maximum 2 sec) after the drop occurred. If the transfer takes longer than 2 sec, the pointer changes to a “ busy” shape whenever it is on the object where the transfer is occurring, and a working message window is displayed and its contents updated every 2-3 sec until the transfer is completed.

3.5.2 Clipboard Transfer

Clipboard transfer allows users to move, copy, and link objects by transferring them from their current location to a temporary clipboard and then from the clipboard to a new location. Clipboard transfer can be used to transfer text or graphics within a window and from one window to another in the same or different applications. A clipboard move consists of cut and paste actions, while a clipboard copy and a clipboard link consists of copy and paste actions.

The application supports clipboard transfer operations. These operations can be executed from a pull-down or pop-up menu or using accelerators. If the application provides access to clipboard transfer in an Edit menu, the Cut, Copy, and Paste options in the menu execute the actions as defined in appendix C. In addition, the accelerators listed in this appendix are available for executing these actions from the keyboard whenever an object that can be edited has keyboard focus.

If the clipboard contents are text, the paste action copies the contents to the location of the text cursor, and any existing text appears to the left of the cursor. If the text has been previously selected when a paste action occurs, the selected text is deleted and the clipboard contents pasted at that location. If the clipboard contents are graphic, the paste action copies the contents to the pointer location in the window with input focus. If the paste action is invoked from the pop-up menu for an object, the clipboard contents are pasted at the insertion point in the object. If the menu is popped up over a selection, the selection is deleted and the clipboard contents replace it if possible. Pasting an object from the clipboard does not select the object. The object that was pasted remains in the clipboard until another object is cut or copied into it. Users are able to view the contents of the clipboard and are informed (e.g., in a message window) when they attempt to cut or copy an object whose size exceeds clipboard capacity.

3.5.3 Primary Transfer (Motif Only)

Primary transfer allows users to transfer a selectable object (e.g., editable text) directly to a destination without using the clipboard for intermediate storage. Primary transfer is performed in the following manner:

> To perform a primary move, users select an object, move the pointer to the destination, and press <Shift> and click BTransfer. A copy of the object is pasted at the location of the pointer, and the original object is deleted.
>
> To perform a primary copy, users select an object, move the pointer to the destination, and press <Ctrl> and click BTransfer. A copy of the object is pasted at the location of the pointer, and the original object remains at its previous location.
>
> To perform a primary link, users select an object, move the pointer to the destination, and press <Ctrl><Shift> and click BTransfer. A link is created from the location of the pointer to the object at its original location.

In addition, if access to primary transfer is provided in an Edit menu, the Primary Copy and Primary Move options execute the actions and use the accelerators defined in appendix C.

Transferring an object by performing a primary copy or a primary link does not select the object; however, transferring an object via a primary move does. When BTransfer is used to perform a primary transfer, the default is a copy operation.

3.5.4 Quick Transfer (Motif Only)

Quick transfer allows users to temporarily select an object and immediately transfer it to a new location. Whereas primary transfer is available for editable objects, quick transfer can be used to transfer static text (e.g., the label for a text field or a push button) or graphics that are not normally selectable. Quick transfer is available in text components in the application and performed in the following manner:

> To perform a quick move, users place the pointer on an object, press <Alt><Shift>, and drag the object using BTransfer. The object is temporarily selected and, when the button is released, moved to the new location.

> To perform a quick copy, users place the pointer on an object, press <Alt><Ctrl>, and drag the object using BTransfer. The object is temporarily selected and, when the button is released, copied to the new location.
>
> To perform a quick link, users place the pointer on an object, press <Alt><Ctrl><Shift>, and drag the object using BTransfer. The object is temporarily selected and, when the button is released, linked to the new location.

When <Alt> and BTransfer are used to perform a quick transfer, the default is a copy operation. When a quick transfer is performed, the object being transferred is not selected when it is displayed at the destination location.

3.6 INTERACTIVE CONTROL

3.6.1 Object-Action Selection Model

User interactions with the application are based on an object-action selection model in which users first select an object (so that it has keyboard focus) and then perform an action on that object. More than one object may be selected, and the objects may be controls (e.g., radio or check buttons), text (e.g., individual characters, strings of characters), or graphic (e.g., track symbols on a map). When users make or change a selection, no other action is performed on the selected objects. Users are informed (e.g., with feedback in the message bar of a window) when an application requires interactions that diverge from this model.

3.6.2 User Control of Interaction

The application executes an action only in response to explicit user input. Users control the pace of the interaction with the application and are not forced to interact with the application at a specified rate. However, the pace of user input does not slow down the speed of application processing or execution.

3.6.3 Immediate Feedback

Users receive an immediate indication that their action has been accepted by the application. Some visible response (e.g., a button is highlighted, the pointer changes shape) is made within 0.2 sec of any user action. When a user's request takes longer than 2-3 sec but less than 10 sec to process, the application changes the pointer shape to "busy" (e.g., a watch or hourglass). When a user's request takes longer than 10 sec to process, the application displays a working message window.[7] Error feedback (e.g., when a user attempts an action that is invalid) is provided to users within 2 sec of the time the error was detected. These guidelines should be considered as recommended response times; it is expected that actual response times will be determined by factors such as the hardware configuration being used, the size of the track database being maintained, and the amount of communications processing being performed.

Users are informed when they can and cannot take an action. Visual cues are provided indicating when the application can accept input (e.g., the pointer appears as a shape supporting selection), when it is temporarily unavailable (e.g., the pointer appears as a watch or hourglass), and when it is unavailable during extended processing (e.g., a working window appears on the screen). In addition, the appearance of the objects displayed by the application provides an indication of their availability. For example, menu options provide feedback concerning the actions that can be performed on selected object(s). If a menu option is not appropriate for an object that has been selected or if a menu option

[7] CDE recommends that a progress indicator be displayed after 10 sec; Windows recommends the indicator be displayed after 5 sec.

cannot be executed for some reason, users are unable to select the menu item and a visual cue (such as graying the option) is provided.

When an operation requires several actions to complete, users are prompted with the actions that need to be performed. For example, when users select a Zoom Area option from a pull-down menu, the application provides feedback explaining how to select the area to be zoomed (e.g., provides a message at the bottom of the window).

The application ignores user actions made during periods when input cannot be accepted. For example, the application accepts keystrokes made by users only when the text cursor appears in the window with input focus; if the cursor is not present in the window, the keystrokes are ignored by the application. The application disables the pointing device and/or keyboard when input may have destructive effects (e.g., when the watch or caution pointer is displayed, indicating that the application is temporarily unavailable or that input cannot be accepted at the location of the pointer). Disabling is particularly important so that any input made by users is not stored and then executed when the application becomes available again. Although an application does not allow users to override disabling, users are able to stop a process if desired (e.g., by activating a Cancel push button).

3.6.4 Error Detection

If users attempt to execute an invalid action, the application does not execute the action except to display a message describing the action that is invalid. If users make multiple errors within a single action, they are notified of each occurrence of an error. The feedback is immediate (i.e., occurs within two seconds of error detection), is visual and/or auditory, and explains the nature of the error made. If users repeat an error, the feedback is different (e.g., auditory with diagnostics or help also provided) to show users that their attempted corrective action was, in fact, processed. Users are required to correct only the invalid action and not to repeat the entire action sequence. After correcting the error, users can execute the same action (e.g., activate a push button) for re-entry that was used for the original entry.

3.6.5 Explicit Destruction

Users are required to confirm a destructive action (i.e., an action with irreversible negative consequences such as deleting a data file) before the action is executed by the application. Users do not have to confirm the action when they close a window unless the action will cause significant data loss. In general, closing a primary window is a potentially destructive action because this type of window usually generates data that should not be lost as a result of such an action. Closing a dialog window is usually nondestructive because these windows are used for obtaining additional user input that does not have to be saved.

When a confirmation prompt is displayed, the window in which the destructive action was taken remains displayed until users make a selection to confirm the action. The window is not removed when the prompt is presented and then redisplayed if users fail to confirm the destructive action. If a destructive action applies to more than one object (e.g., multiple files), users are provided with a list of the objects and allowed to select the one(s) to which the action should be applied.

3.6.6 General Undo Capability

Users are able to undo the action or selection most recently made unless the selection was one requiring explicit destruction. For example, users are able to undo actions such as cut, copy, and paste, deselect objects, return an object to its prior state before an action was executed, and retrieve information that was removed from the screen. If an undo capability cannot be provided, the application labels irreversible actions as such and clearly separates actions that are reversible from those that are not.

3.6.7 Use of Processing Modes

Processing modes are states where user actions have different results depending on the mode currently in effect. Both Motif and Windows use modes when defining how users interact with some elements of the user interface. For example, the mode assigned to a dialog window determines the extent to which users can interact with other windows while the dialog window is displayed (see section 4.1.2.2). The application is designed so that the same action has the same effect whenever it is executed by users. If an application-specific processing mode is required for a particular function, users are provided with a visual cue to indicate the mode currently in effect. For example, when users select a drawing tool from a toolbar, the tool button remains highlighted to indicate the mode in which users are operating.

3.6.8 Consistency in Performing Operations

Developers need to identify the set of operations (e.g., updating data records, building map overlays) that are supported within the application and then specify the sequence of actions that users follow to perform each operation. This sequence allows users to complete the operation rapidly and efficiently (i.e., minimizes the number of keystrokes executed and windows opened). For example, if an application requires users to enter location information for a contact by clicking the position on a map, the application records the position and uses the information; users do not have to type the same position information in a later part of its dialog with the application. If the application does not record the position selected, then users do not have to perform the action (i.e., click on a position). Users are able to execute the same sequence of actions whenever they perform the same operation in the application.